REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
		Final Technical Report

4. TITLE AND SUBTITLE

Wave Evolution and Laminar-Turbulent Transition in Fully 3D Supersonic Flows

5. FUNDING NUMBERS

Contract
F49620-95-C-0003

6. AUTHOR(S)

Thorwald Herbert

AFOSR-TR-96

0293

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

DynaFlow, Inc.
P.O. Box 21319
3040 Riverside Dr.
Columbus, OH 43221

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Office of Scientific Research NA
110 Duncan Avenue
Bolling AFB DC 20332-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

95-C-0003

11. SUPPLEMENTARY NOTES

19960618 016

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. ABSTRACT (Maximum 200 words)

The analysis of transition in fully three-dimensional boundary layers on high-speed aircraft is hampered by the lack of a rigorous theory of instability in these flows and the lack of accurate data to evaluate approximations. To provide such data, we have in this first phase of a longer-term effort designed a computational framework for this task. This framework allows interactive computations and data analysis in connected blocks of the large physical domain on networks of computers. Theoretical studies on the stability of three-dimensional boundary layers have been performed to obtain guidance on the computational treatment of the problem. A multi-grid method specially adapted to the analysis of wave propagation in shallow domains has been developed and implemented. Testing of the code could not be completed before the project was terminated.

14. SUBJECT TERMS	15. NUMBER OF PAGES	16. PRICE CODE
	11	

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unlimited	SAR	SAR	SAR

DTIC QUALITY INSPECTED 1

Wave Evolution and Laminar-Turbulent Transition in Fully 3D Supersonic Flows

Thorwald Herbert
DynaFlow, Inc.
P.O. Box 21319
Columbus, OH 43221, USA

April 10, 1996

FINAL REPORT
Air Force Office of Scientific Research
Contract F49620-95-C-0003

Abstract

The analysis of transition in fully three-dimensional boundary layers on high-speed aircraft is hampered by the lack of a rigorous theory of instability in these flows and the lack of accurate data to evaluate approximations. To provide such data, we have in this first phase of a longer-term effort designed a computational framework for this task. This framework allows interactive computations and data analysis in connected blocks of the large physical domain on networks of computers. Theoretical studies on the stability of three-dimensional boundary layers have been performed to obtain guidance on the computational treatment of the problem. A multi-grid method specially adapted to the analysis of wave propagation in shallow domains has been developed and implemented. Testing of the code could not be completed before the project was terminated.

Contents

1 Overview

This work aimed at developing and applying analytical and computational methods to investigate the propagation of instability waves in steady, compressible, three-dimensional laminar boundary-layers. The first method relies on a 3D extension of the Parabolized Stability Equations (PSE) to account for nonlocal effects due to flow variations tangent to the vehicle surface and predict the associated modulations of the disturbance amplitude and wavenumber. The second method is based on a time-accurate Navier-Stokes solver (DNS) and is considered the long-term solution of the problem, though still expensive today. To enable efficient applications and future extensions, the code has been designed for distributed execution, and interactive analysis and documentation of the dynamic motion during execution. Guided by theoretical studies and experimentation with prototypes, the numerical multigrid method has been carefully adapted to the problem of wave propagation in shallow domains.

Owing to budget cuts, the work on the PSE approach has been postponed in favor of the DNS code. In a longer-term effort, the work aimed at using this Navier-Stokes code to compute the steady, laminar basic flow, to analyze the propagation of linear waves, and to predict their nonlinear evolution toward transition. The results were to be compared with predictions from local linear stability theory and PSE analysis to quantify the the effects of three-dimensionality and to improve methods for transition analysis in fully three-dimensional boundary layers for practical applications. These goals could not be achieved because of early termination of the project.

2 Achievements

Intense efforts have been spent on the design and development of prototypes for the modular C++ code. A framework for interactive computations in multiple blocks on distributed computers has been developed. The framework is independent of the physical problem and the numerical method. This framework is operational, including the distributed execution of computational blocks, their communication, and interactive data collection for display in different forms. Communication is based on `PVM` and operational on our heterogeneous network of workstations (SGI, HP, IBM). The code can be executed interactively or in batch mode, using an easy-to-read command script. The script can be prepared in interactive mode with suppressed execution. The code also performs intermediate static, dynamic, or post processing.

A theoretical study on the stability of fully 3D boundary layers has been completed to obtain guidance on the computational treatment of the prob-

lem. A multigrid code has been written to solve the incompressible Navier-Stokes equations to allow faster debugging and testing as well as comparison with readily available data. Numerous improvements in algorithms and implementation of the multigrid method have been made. The code is near the end of the debugging phase. In parallel, we have developed suitable formulations for the compressible problem and analyzed different methods of implementing general curvilinear coordinates for efficiency with respect to speed and memory.

3 Accomplishments

Theoretical studies have been performed on the linear stability of fully 3D boundary layers. This subject has been virtually ignored in the literature except for the development of some heuristic concepts used under the parallel-flow assumption. Our analysis has shown that these concepts, which were "imported" from conservation laws for non-dissipative systems, are unjustified. Without these conservation properties, the propagation of linear and nonlinear instability waves in 3D boundary layers cannot be traced by solving initial-boundary-value problems along lines, but requires initial data given in a plane.

The first method envisioned to trace the evolution of the field by marching in planes downstream is based on an extension of the PSE method. Iteratively solving for the solution in each plane at every streamwise marching step is more costly than our existing PSE approach for quasi-3D boundary layers, but less costly than DNS because there is no feedback (and no problem with outflow boundaries). However, the development of this technique including proper numerical methods is more demanding than building on existing techniques for DNS. In view of the reduced budget at the time of contract award, and expected further cuts for the next fiscal year, we have decided to delay working on the PSE approach in favor of completing the second method discussed in the following.

3.1 Unsteady Field Analysis

We have focused on the second approach which rests on a time-accurate solver for Navier-Stokes equations and (nonlinear) stability equations for compressible boundary-layer flows in curvilinear coordinates. This approach was suggested by our experience (Mack & Herbert 1995) with a code by Liu & Liu (1993) and Liu et al. (1993) to simulate transition in an incompressible flat-plate boundary layer. Our work with a modification of this code on active flow control problems (AFOSR Contract F49620-93-1-0135) revealed two important facts: (1) the impressive efficiency of the multigrid method

implemented in this code, and (2) major shortcomings besides the restriction to incompressible flow in Cartesian coordinates. Some of these shortcomings are: the need to run the code on a single computer (preferably a remote Cray), the lack of any interactivity, the need to transfer large data files to local computers for visual data analysis, and the complete reliance on post-processing after the code has progressed to some time specified at the start of the run. This mode of operation has been acceptable for steady problems, but is inappropriate for the unsteady research tasks at hand. The separation of data production and analysis requires to either store or discard the produced data without knowing the result. Recording the results of unsteady computations on large grids exceeds the capacity of contemporary storage media.

Computations of unsteady motions require the capability to analyze the data during production. Since computers for data analysis have different characteristics (high-level graphics, visualization software) than those for raw computation (high speed, large memory), provisions have to be made to perform the unsteady field analysis simultaneously on different computers. Once this necessary step is taken, it is only a logical generalization to distribute the data production over a network of computers to pool their speed, memory, and storage capacity. This larger pool of resources is necessary to extend the analysis to compressible flows and/or curvilinear coordinates.

We have completed the design and implementation of these concepts in a modular framework that is independent of the physical problem to be solved and the numerical method used. An overview of the code is shown in figure 1.

Each of the modules (shown as rectangles) is an executable object code and thus can run on some assigned computer of a given network. The modules named `Block n` perform the actual computation in a part of the physical domain, and may be identical or different. The number of blocks is independent of the number of computers in the network, which allows tests of large problems on a small network or even on a single computer. The communication between modules is based on `PVM`, a publicly available and widely used software for distributed processing. The multiple (and virtually redundant) data connections permit to run various combinations of modules in different operational modes. The combination of `Pilot, Hub, Block 1, ..., Block N` can run as a batch job driven by a prepared command file. This combination, together with `Display` can run interactively, using a command file and additional instructions from the keyboard. However, it is more convenient, and less prone to errors, to operate the code interactively through a graphical user interface (`GUI`). The combination of `GUI, Display` and `Reader` can be regarded as a traditional visualization code for the analysis of static data. Access to the distributed results of the computation for the complete physical domain is through the `Hub` that coordinates the blocks

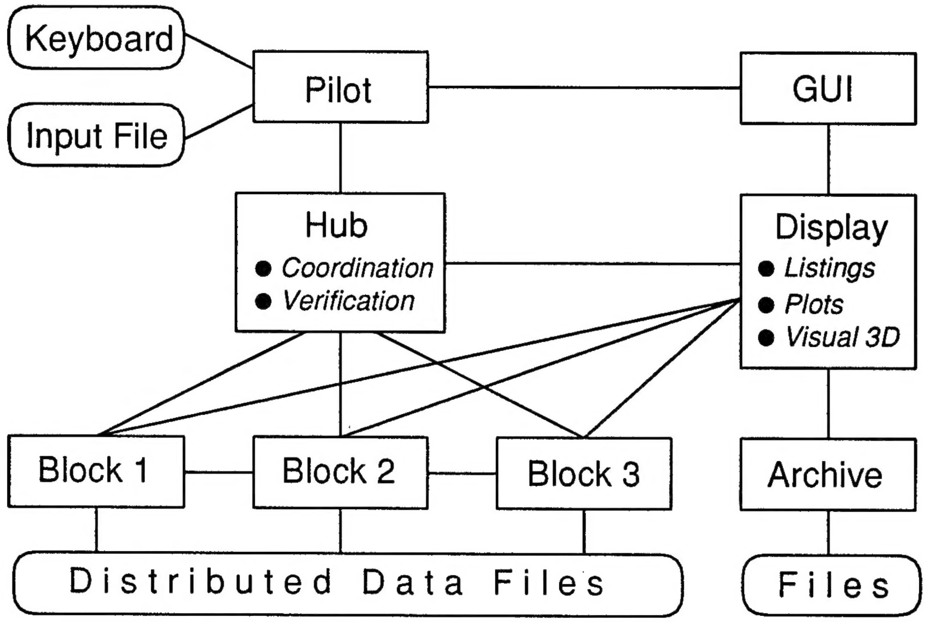

Figure 1: Overview of the modules and data flow of the code `UFA` for unsteady field analysis.

as a group with all other components. The `Hub` also verifies the consistency of input data, integrity of the physical domain, and proper performance of the network communication. The `GUI` provides status information of the computation, and drives `Pilot` and/or `Display` using a common command language. The commands are saved in a command (or script) file and can be rerun as input file. With the execution suppressed, the `GUI` can be used to prepare command files for batch or unattended interactive execution. The latter mode of operation is useful e.g. for overnight video recording of the computational results.

The module `Display` currently performs listings or communicates with the `Archive` to store selected data in files. Instead of writing software for plotting and visualization, we use the publicly available software `XMGR` for plotting, and `Visual3` for visual data analysis. The latter choice was made for two reasons: `Visual3` consists of a set of subroutines that can be called by `Display`, and is part of `pV3` (publicly available). `pV3` is a distributed visualization software that communicates with clients via `PVM`, and can be integrated into our application.

Except for the `GUI` and `pV3`, all modules are fully operational. The overhead caused by distribution of the tasks has been measured and is negligible in comparison with the time of the computation. Only the startup of the modules on remote computers causes a sluggish start of the overall code. The existing modules allow execution of the computational blocks simultaneously

with display of the data. This capability is crucial for the efficient debugging and testing of the computational blocks. The difficulty of debugging distributed codes is considerable.

3.2 Multigrid Method

A careful analysis of the multigrid technique used by Liu et al. in the light of our computations with a modified version of their code suggests some changes in this technique. Their multigrid acceleration uses partial coarsening in the y, z plane normal to the flow direction, and Gauss-Seidel iteration to solve the momentum equations along lines in the y direction normal to the wall. This strategy has two consequences: a poor distribution of the residuals in the streamwise x direction, and a severe restriction on the time step to achieve diagonal dominance of the tridiagonal systems, which is necessary for the convergence of the Gauss-Seidel iteration. In contrast to their procedure, we use partial coarsening in the x, z plane which is more efficient because of the large domain size in these directions, and more efficiently distributes and corrects residuals in these directions. We also replace the Gauss-Seidel iteration by a direct solution of the tridiagonal systems. While this step about doubles the time needed to solve the system, it relaxes the restriction on the time step, and we expect higher overall performance. In addition, the Thomas algorithm balances the residuals in the remaining y direction, i.e. between the multi-gridded planes parallel to the boundary. A more satisfactory and highly efficient algorithm has also been developed to compute the pressure from the continuity equation.

One of the greatest hassles of multigrid codes is the bookkeeping of indices and access to the correct data on the staggered grids for each variable on different grid levels. Together with the different treatment in the interior and near boundaries (and in our case, the decomposition of the physical domain), the numerous case distinctions lengthen the programs, and the extensive indexing and index substitutions make even the code of Liu et al. for a relatively simple Navier-Stokes problem unreadable and unmaintainable for practical purpose. We have proof of this statement because we corrected numerous errors in this "commercial" code. In response to this problem area, we have developed an elegant procedure to implement equations in symbolic variables, which are defined by appropriate `#define` statements in C++. Structures that are set at startup provide all needed data at different levels. Pointers are used to distinguish the finite-difference formulas at interior or boundary points. These concepts cannot be exploited in Fortran. As a consequence of these efforts, the lengths of the subroutines shrank to a small fraction of those in the Fortran code of Liu et al., a tremendous advantage for debugging. Much of the index calculations is done in the header files

included in every routine.

Additional considerations were necessary to decide on the strategies for the decomposed physical domain. The overhead for data exchange between blocks can be reduced by restricting this exchange to the finest grid level. In this case, the number of multigrid levels is local to each block, but the solution is not completely independent of the specific decomposition. To make the solution completely independent of the domain decomposition, data have to be exchanged on all levels, and the lowest multigrid levels in all blocks must agree. In spite of this restriction, we have decided in favor of this second option.

3.3 Computational Codes

With the framework in place, we have pursued two parallel developments, the first of which serves to establish full functionality of the multigrid method and the data communication within and across blocks. For this task, we have chosen the incompressible Navier-Stokes equations with a rather general set of boundary conditions, and the associated nonlinear inhomogeneous disturbance equations. Plane Poiseuille flow and Blasius boundary layer are ideal test cases because data for comparison are readily available and the testing requires a fraction of the time needed for the compressible problem. This code is completely written and formally runs over the specified number of time steps. Two types of blocks are used for the physical field and the buffer zone at outflow boundaries, respectively. The testing of the code has made good progress and was almost complete at the time of project termination.

We will complete and maintain this incompressible code as a valuable prototype that can serve to answer numerous open questions for the wider class of quasi-three-dimensional Falkner-Skan flows, such as the breakdown of crossflow vortices. This code can also serve to implement general curvilinear coordinates and to test the efficiency of different concepts. With this extension, this by-product of the overall effort can be used for a wider class of problems, e.g. for a preliminary analysis of the fully 3D boundary layer over the swept wing of commercial (subsonic) airplanes, especially in the neighborhood of the engines. The results of such an analysis would be valuable to support the LFC efforts at Boeing.

Leaning on the incompressible prototype, the second line of development concerns efficient formulations of the compressible equations and the aspects of curvilinear coordinates. For a good part of this effort, we have taken advantage of earlier results for DynaFlow's PSE code (Stuckert et al. 1995). Different sets of equations have been generated using Mathematica for symbolic manipulations. The equations are automatically coded in a form consistent with the prototype. Using a second-order finite-difference method, the

equations have been tested for a single block without multigrid acceleration. After optimizing compilation, the speed and memory demand of a given set of equations can be evaluated. This selection process for the implementation of compressibility has made good progress, and similar comparisons for the implementation of curvilinear coordinates have been performed. The effort to establish conclusive data on time and memory demand was interrupted by the termination of the project.

Since the prototype has been designed to easily adapt to different physical problems or differential equations, the following three codes would have been basically functional soon after the testing of the current prototype was completed: the incompressible version in curvilinear coordinates, the compressible version in stretched Cartesian coordinates, and the compressible version in curvilinear coordinates. We hope for a future opportunity to complete these codes and to enter the phase of data production. We are confident that the work during this first phase has provided an excellent basis for computational analysis of unsteady phenomena in general.

4 Personnel Supported

The following personnel has participated in the work and has been partially supported under this contract:

Thorwald Herbert, Principal Investigator

Shiling Huang, Research Scientist

Charlotte Herbert, Systems Programmer

5 Publications

NONE. We have written code design documents which we intend to keep proprietary. We will use the incompressible version of the code to produce data for publication. We had planned a first report on the capabilities of the code in the proceedings of the *First AFOSR Conference on Dynamic Motion CFD*, June 3-5, 1996, New Brunswick, New Jersey. This plan was cancelled. Some theoretical results are included in a contribution on "Parabolized Stability Equations" to *Annual Reviews in Fluid Mechanics*, Vol. 29, 1997.

5.1 Participation/Presentations at Meetings

NONE. We had planned a code demonstration by computer or video at the *First AFOSR Conference on Dynamic Motion CFD*, June 3-5, 1996, New

Brunswick, New Jersey, and in an invited lecture at the *27th AIAA Fluid Dynamics Meeting*, June 17-20, New Orleans, Louisiana. These plans were cancelled.

5.2 Transitions

Dr. J. D. Crouch, Boeing Commercial Airplane Group, has expressed strong interest of the Computational Fluid Dynamics Lab and the Laminar Flow Control Branch in our development of this software. Dr. Crouch specifically addressed the stability problems in those sections of the swept wings where the flow is modified by pylons and engines. These flows cannot be reliably analyzed with current tools, hampering LFC efforts. We have discussed details of a future cooperation on these problems.

5.3 Honors/Awards

Thorwald Herbert was elected Fellow of the American Physical Society in 1987. He is an Associate Fellow of the American Institute of Aeronautics and Astronautics since 1993.

References

[1] Mack, L. M. and Herbert, Th. 1995 "Linear Wave Motion from Concentrated Harmonic Sources in Blasius Flow," AIAA 95-0774.

[2] Liu, C. and Liu, Z., 1993 "Multigrid Methods and High-Order Finite-Difference for Flow in Transition," AIAA 93-3354.

[3] Liu, C., Liu, Z., and McCormick, S. M. 1993 "Multigrid Methods for Flow Transition in Three-Dimensional Boundary Layers with Surface Roughness," NASA Contractor Report 4540.

[4] Stuckert, G. K., Lin, N., and Herbert, Th. 1995 "Nonparallel Effects in Hypersonic Boundary Layer Stability," AIAA 95-0776.